

EXAMINER'S AMENDMENT

1. An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it **MUST** be submitted no later than the payment of the issue fee.

Authorization for this examiner's amendment was given in a telephone interview with Hussein Akhavannik on July 30, 2008.

The application has been amended as follows:

Amendments to the Specification

Please amend the title, as shown below:

Self-Contained Applications that are Applied to be Received by and Processed Within a Browser Environment and that have a First Package that Includes a Manifest File and an Archive of Files Including a Markup Language File and a Second Package

Please replace the Abstract, with the following Abstract:

An implementation of a self-contained application may be distributed and automatically executed within a network client environment, such as within a web browser environment. The implementation of the self-contained application includes a first package that includes a manifest and an archive of files. The archive of files includes a markup language file that is the initial file to be processed and that includes instructions for initiating execution of the application. In the implementation, the archive of files also includes other files needed to execute the application and a second package that may be of the same type as the first package. The manifest file includes an initial file identifier that instructs the application to process the markup language initial file before processing the other files.

Amendments to the Claims

This listing of claims replaces all prior versions and listings of claims in the application.

Listing of Claims:

1-32. (Cancelled)

33. (Currently Amended) A method for executing an application, the method comprising:
loading a first package within a browser on a local client computer, the first package
including a first manifest and a first archive of files that includes instructions and content needed
to execute an application, the first archive of files including:

an initial file that includes instructions for initiating execution of the
application, the initial file being a markup language file,

other files needed to execute the application, and

a second package that includes a second manifest and a second archive of
files, the second package being of the same type as the first package, and the
second archive of files being of the same type of file structure as the first archive
of files;

the first manifest including an initial file identifier that indicates that the initial
file is to be processed before the other files in the first archive of files when the
application is executed;

in response to loading the first package within the browser, automatically accessing the first manifest;

locating the initial file identifier in the first manifest;

based on the located initial file identifier, accessing the instructions for initiating the execution of the application from the initial file;

processing the accessed instructions from the initial file; and

automatically initiating execution of the application based on the processed instructions.

34. (Previously Presented) The method of claim 33, further comprising receiving the first package at the local client computer.

35. (Previously Presented) The method of claim 34, wherein receiving the first package at the local client computer comprises loading the first package onto the local client computer from a local computer readable medium.

36. (Previously Presented) The method of claim 33, wherein the instructions for initiating execution of the application from the initial file include instructions to process additional instructions from one or more of the other files in the first archive, the method further comprising:

accessing the additional instructions from the one or more other files; and

processing the additional instructions, the accessing and processing of the additional instructions being performed in response to processing the instructions for initiating execution of the application from the initial file.

37. (Previously Presented) The method of claim 33, wherein:
the initial file comprises a source file for a web page,
the instructions for initiating execution of the application from the initial file comprise instructions for rendering the web page, and
automatically initiating execution of the application comprises rendering the web page in accordance with the instructions for rendering the web page.

38. (Previously Presented) The method of claim 37, wherein the source file for the web page comprises an HTML document.

39. (Previously Presented) The method of claim 33, wherein:
the initial file comprises an executable file, and
the instructions for initiating execution of the application from the initial file comprise program execution instructions.

40-41. (Cancelled)

42. (Previously Presented) The method of claim 33, wherein:
the first manifest further comprises an archive type identifier that identifies an application type of the application, and
processing the accessed instructions comprises processing the accessed instructions in accordance with the application type of the application.

43. (Previously Presented) The method of claim 33, further comprising displaying an icon associated with the application, wherein loading the first package in the browser comprises automatically loading the first package in the browser in response to a user selecting the icon associated with the application.

44. (Previously Presented) The method of claim 33, wherein the initial file identifier indicates that the initial file is the first file to be processed in the first archive of files when the application is executed.

45. (Previously Presented) The method claim 33, wherein automatically initiating execution of the application based on the processed instructions comprises automatically initiating execution of the application without maintaining a connection between the local client computer and a web server.

46. (Currently Amended) A computer program product for executing an application , the computer program product being embodied in a tangible computer readable medium and including instructions that, when executed by a processor, cause the processor to:

load a first package within the computer program product, the first package including a first manifest and a first archive of files that include instructions and content needed to execute the application, the first archive of files including:

an initial file that includes instructions for initiating execution of the application, the initial file being a markup language file,
other files needed to execute the application, and

a second package that includes a second manifest and a second archive of files, the second package being of the same type as the first package, and the second archive of files being of the same type of file structure as the first archive of files;

the first manifest including an initial file identifier that indicates that the initial file is to be processed before the other files in the first archive of files when the application is executed;

automatically access the first manifest in response to loading the first package within the computer program product;

locate the initial file identifier in the first manifest;

access the instructions for initiating the execution of the application from the initial file based on the located initial file identifier;

process the accessed instructions from the initial file; and

automatically initiate execution of the application based on the processed instructions.

47. (Previously Presented) The computer program product of claim 46, further comprising instructions that, when executed by the processor, cause the processor to receive the first package at a local client computer.

48. (Previously Presented) The computer program product of claim 47, wherein the instructions that, when executed by the processor, cause the processor to receive the first package at the local client computer comprise instructions that, when executed by the processor, cause the

processor to load the first package onto the local client computer from a local computer readable medium.

49. (Previously Presented) The computer program product of claim 46, wherein the instructions for initiating the execution of the application from the initial file include instructions to process additional instructions from one or more of the other files in the first archive, the computer program product further comprising instructions that, when executed by the processor, cause the processor to:

access and process the additional instructions from the one or more other files in response to processing the instructions for initiating execution of the application from the initial file.

50. (Previously Presented) The computer program product of claim 46, wherein:
the initial file comprises a source file for a web page,
the instructions for initiating execution of the application from the initial file comprise instructions for rendering the web page, and

the instructions that, when executed by the processor, cause the processor to automatically initiate execution of the application comprises instructions that, when executed by the processor, cause the processor to render the web page in accordance with the instructions for rendering the web page.

51. (Previously Presented) The computer program product of claim 50 wherein the source file for the web page comprises an HTML document.

52. (Previously Presented) The computer program product of claim 46, wherein:
the initial file comprises an executable file, and
the instructions for initiating execution of the application from the initial file comprise program execution instructions.

53-54. (Cancelled)

55. (Previously Presented) The computer program product of claim 52, wherein:
the first manifest further comprises an archive type identifier that identifies an application type of the application, and
the instructions that, when executed by the processor, cause the processor to process the accessed instructions comprise instructions that, when accessed by the processor, cause the processor to process the accessed instructions in accordance with the application type of the application.

56. (Previously Presented) The computer program product of claim 46 further comprising instructions that, when executed by a processor, cause the processor to display an icon associated with the application, wherein the instructions that, when executed by the processor, cause the processor to load the first package in the computer program product comprise instructions that, when executed by the processor, cause the processor to automatically load the first package in the computer program product in response to a user selecting the icon associated with the application.

57. (Previously Presented) The computer program product of claim 46, wherein the computer program product comprises a browser.

58. (Previously Presented) The computer program product of claim 46, wherein the initial file identifier indicates that the initial file is the first file to be processed in the first archive of files when the application is executed.

59. (Previously Presented) The computer program product of claim 46, wherein the instructions that, when executed by the processor, cause the processor to automatically initiate execution of the application based on the processed instructions comprise instructions that, when executed by the processor, cause the processor to automatically initiate execution of the application without maintaining a connection between a local client computer and a web server.

60. (Currently Amended) A method for generating a first package such that an application may be automatically executed by a browser on a client computer, the method comprising:

generating a first archive of files that include instructions and content needed to execute the application, the first archive including:

an initial file that includes instructions for initiating execution of the application,
the initial file being a markup language file,
other files needed to execute the application, and

a second package that includes a second manifest and a second archive of files,
the second package being of the same type as the first package, and the second archive of
files being of the same type of file structure as the first archive of files;
generating a first manifest file that is associated with the first archive, the first manifest
file including an initial file identifier that instructs the browser to process the initial file before
processing other files in the first archive in order to initiate execution of the application; and
encapsulating the first archive of files and the first manifest file within the first package.

61. (Previously Presented) The method of claim 60, wherein the instructions for initiating
execution of the application from the initial file include instructions to process additional
instructions from one or more of the other files in the first archive.

62. (Previously Presented) The method of claim 60, wherein:
the initial file comprises a source file for a web page, and
the instructions for initiating execution of the application from the initial file comprise
instructions for rendering the web page.

63. (Previously Presented) The method of claim 62, wherein the source file for the web
page comprises an HTML document.

64. (Previously Presented) The method of claim 60, wherein:
the initial file comprises an executable file, and

the instructions for initiating execution of the application from the initial file comprise program execution instructions.

65-66. (Cancelled)

67. (Previously Presented) The method of claim 60, wherein:

the first manifest file further comprises an archive type identifier that instructs the browser to process the instructions for initiating execution of the application from the initial file in accordance with the application type of the application.

68. (Currently Amended) A self-contained package, the self-contained package being embodied in a tangible computer readable medium and configured to enable an application to be automatically executed within a browser environment, the self-contained package comprising:

a first archive including:

an initial content source having instructions for initiating execution of the application within the browser environment, the initial content source being a markup language file,

additional files that include instructions and content needed to execute the application within the browser environment, and

a second package that includes a second manifest and a second archive of files,
the second package being of the same type as the first package, and the second archive of files being of the same type of file structure as the first archive of files; and

a first manifest associated with the first archive that includes an initial content identifier that indicates that the initial content source is to be processed before the additional files in the first archive when the application is executed.

69. (Previously Presented) The self-contained package according to claim 68, wherein said initial content source is an HTML file containing content layout instructions for rendering a document that displays content included in the first archive.

70. (Previously presented) The self-contained package according to claim 68, wherein said initial content source is an executable file containing program execution instructions.

71. (Cancelled)

72. (Previously Presented) The self-contained package of claim 70, wherein the first manifest further comprises an archive type identifier that identifies an application type of the application.

73. (Cancelled)

74. (Previously Presented) The self-contained package of claim 68, wherein the additional files comprise one or more of a web page, a script, an image, a sound file, and a JAVA file.

75-92. (Cancelled)

93. (Previously Presented) The method of claim 34, wherein receiving the first package at the local client computer comprises receiving an e-mail that includes the first package at the local client computer.

94. (Previously Presented) The method of claim 60 further comprising, establishing security credentials for the initial file, the other files needed to execute the application, and the second archive using a single signature for the first package.

95. (Cancelled)

96. (Previously Presented) A method for executing an application, the method comprising:

loading, within a processing environment on a local client computer, a first package that includes a first manifest and a first archive of files, wherein:

the first manifest includes a first initial file identifier that identifies a particular file from within the first archive of files as a first initial file that is to be processed before other files in the first archive of files when the application is executed, and

the first archive of files includes:

the first initial file, the first initial file including instructions for initiating execution of the application, the first initial file being a markup language file,
other files needed to execute the application, and
a second package, the second package including:

a second manifest that includes a second initial file identifier that identifies a particular file from within a second archive of files as a second initial file that is to be processed before other files in the second archive of files when the application is executed, and

the second archive of files that includes:

the second initial file, the second initial file including instructions for executing the application, the second initial file being a markup language file, and

additional files needed to execute the application;

in response to loading the first package within the processing environment, automatically accessing the first manifest;

locating the first initial file identifier in the first manifest;

based on the located first initial file identifier, accessing the instructions for initiating execution of the application from the first initial file;

processing the accessed instructions from the first initial file; and

automatically initiating execution of the application based on the processed instructions from the first initial file.

97. (Cancelled)

98. (Previously Presented) The method of claim 96 further comprising:

accessing the second manifest after locating the first initial file identifier in the first;

locating the second initial file identifier in the second manifest;
based on the located second initial file identifier, accessing the instructions for executing the application from the second initial file;
processing the accessed instructions from the second initial file; and
executing the application based on the processed instructions from the second initial file.

99-101. (Cancelled)

102. (New) The method of claim 33, wherein the same type of file structure of the first archive of files and the second archive of files is a .JAR type file structure.

103. (New) The computer program product of claim 46, wherein the same type of file structure of the first archive of files and the second archive of files is a .JAR type file structure.

104. (New) The method of claim 60, wherein the same type of file structure of the first archive of files and the second archive of files is a .JAR type file structure.

105. (New) The self-contained package according to claim 68, wherein the same type of file structure of the first archive of files and the second archive of files is a .JAR type file structure.

Allowable Subject Matter

The following is an examiner's statement of reasons for allowance:

Applicant amended independent claims 33, 46, 60 and 68 to include allowable subject matter. Said claims now specify that there is a first and second package of the same type, each package including a manifest and archive of files, where the first archive of files includes said second package; that is, the second package is nested within the first package's archive. The package structure is utilized as a method of executing applications, and are loaded via a client's web browser. Cited prior art does teach packages, loaded within web browsers, and where packages are utilized for executing applications. As Applicant has argued, the prior art does not, however, teach or suggested the initial file was a markup language file and where the file structure, as disclosed by Applicant is utilized. The prior art, including Java JAR files, does not teach or suggest methods including the file structure described and claimed by Applicant.

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to John M. MacIlwain whose telephone number is (571) 272-9686. The examiner can normally be reached on M-F 7:30AM - 5:00PM EST; off alternate Fridays.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Andrew Caldwell can be reached on (571) 272-3868. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

John MacIlwinen

(571) 272 – 9686

/Andrew Caldwell/
Supervisory Patent Examiner, Art Unit 2142